(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102075434 A
(43)申请公布日 2011.05.25

(21)申请号 201110031925.9

(22)申请日 2011.01.28

(71)申请人 华中科技大学
地址 430074 湖北省武汉市洪山区珞喻路1037号

(72)发明人 金海 吴松 石宣化 付宇

(74)专利代理机构 华中科技大学专利中心 42201
代理人 方放

(51)Int.Cl.
H04L 12/56(2006.01)
H04L 29/08(2006.01)
H04L 1/18(2006.01)

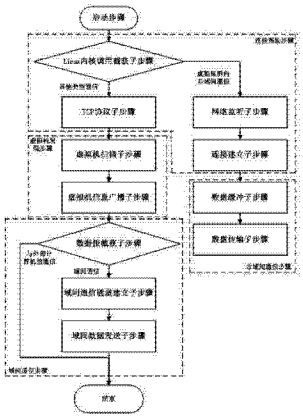
权利要求书 2 页 说明书 6 页 附图 2 页

(54)发明名称

一种虚拟集群中的通信方法

(57)摘要

一种虚拟集群中的通信方法，属于虚拟化及集群计算领域，解决虚拟化技术下的网络通信效率性能低下、非域间通信时虚拟集群中TCP/IP通信协议过于复杂、域间通信时数据报穿越的路径过长导致域间通信性能低下的问题，以提高虚拟集群网络通信效率。本发明包括启动步骤、连接帮助步骤、非域间通信步骤、虚拟机发现步骤以及域间通信步骤。本发明对虚拟集群内非域间通信进行专门设计，可以显著提高网络通信的效率，降低处理器的负担；对域间通信采用在虚拟机之间开辟域间通信通道，大大提高网络通信的性能；本发明具有对已有应用程序的二进制兼容性，不依赖于虚拟集群使用的具体网络硬件，效率高、可用性高。



CN 102075434 A

1. 一种虚拟集群中的通信方法，包括启动步骤、连接帮助步骤、非域间通信步骤、虚拟机发现步骤以及域间通信步骤，其特征在于：

(1) 启动步骤，为每台虚拟机创建虚拟字符设备，在每台虚拟机的内存中设置通道指针数组、窗口值 W 和时间 T，每台虚拟机在所在物理计算机的一块内存 Xen Store 中写入自己的虚拟机 ID 和 MAC 地址信息；所述窗口值 W 为 64 ～ 256 个，时间 T 为 0.1 ～ 1ms；

(2) 连接帮助步骤，包括下述子步骤：

(2.1)Linux 内核调用截获子步骤：请求建立 TCP 连接的虚拟机修改 Linux 内核代码中的函数指针，截获上层应用程序对 TCP 连接建立函数的调用指令，根据调用指令中的目标 IP 地址判断网络通信是否属于虚拟集群内的非域间通信，是则请求建立一个用于建立轻量级通信连接时交换所需信息的 TCP 连接，转子步骤 (2.2)，否则转子步骤 (2.4)；

(2.2) 网络监听子步骤：TCP 连接的目标虚拟机接受 TCP 连接请求，把 TCP 连接套接字描述符写入到自身的虚拟字符设备中，转子步骤 (2.3)；

(2.3) 连接建立子步骤：写入 TCP 连接套接字描述符后，目标虚拟机在自身的内存中分配数据缓冲区并将其初始化，在通道指针数组中为该数据缓冲区分配通道号，然后通过 TCP 连接向请求建立 TCP 连接的虚拟机发送自身的 MAC 地址和通道号，接收对方的 MAC 地址和通道号，完成轻量级协议连接的建立，转步骤 (3)；

(2.4)TCP/IP 协议子步骤：通过 TCP/IP 协议的三次握手过程与目标地址建立起 TCP 连接，在传输数据时，TCP/IP 协议采用超时重传和捎带确认机制保证数据传输的可靠性，在流量控制上，TCP/IP 协议采用滑动窗口的方式，已发送且未确认的数据的字节数最大不能超过滑动窗口的大小，在 TCP/IP 协议完成数据报的处理后，转步骤 (4)；

(3) 非域间通信步骤，包括下述子步骤：

(3.1) 数据缓冲子步骤：请求建立 TCP 连接的虚拟机或者目标虚拟机发送数据时，将要发送的数据存放在自身数据缓冲区的发送缓冲区中，转子步骤 (3.2)；请求建立 TCP 连接的虚拟机或者目标虚拟机接收数据时，将接收到的数据存放在自身数据缓冲区的接收缓冲区中，等待应用程序读取；

(3.2) 数据传输子步骤：从发送缓冲区中取出数据组装成数据报，将数据报通过 Xen 的虚拟网络接口发送给对方虚拟机，直至所要发送的数据全部发送完毕；

(4) 虚拟机发现步骤，包括下述子步骤：

(4.1) 虚拟机扫描子步骤：请求建立 TCP 连接的虚拟机所在的物理计算机周期性地扫描自身内存的 Xen Store 区域，获取参与域间通信的虚拟机列表信息，转子步骤 (4.2)；

(4.2) 虚拟机信息广播子步骤：将参与域间通信的虚拟机列表信息组成列表信息数据报，通过网络广播到每一个参与域间通信的虚拟机中去，转步骤 (5)；

(5) 域间通信步骤，包括下述子步骤：

(5.1) 数据报截获子步骤：请求建立 TCP 连接的虚拟机收到列表信息数据报，将其中的虚拟机列表信息存放在自身内存中；然后截获经过 TCP/IP 协议处理的数据报，查看该数据报的目标地址是否包括在所述虚拟机列表信息中，是则转子步骤 (5.2)，否则将数据报通过 Xen 的虚拟网络接口发送给目标地址；

(5.2) 域间通信通道建立子步骤：请求建立 TCP 连接的虚拟机查看是否和目标虚拟机之间已经建立域间通信通道，是则转子步骤 (5.4)，否则转子步骤 (5.3)；

(5.3) 在自身内存中分配读入缓冲区和写出缓冲区并创建一个事件通道，然后发送通道建立数据报通知目标虚拟机将两个缓冲区的内存地址映射到目标虚拟机的内存地址空间中并且与所创建的事件通道绑定，目标虚拟机收到通道建立数据报后完成所述映射和绑定操作，转子步骤 (5.4)；

(5.4) 域间数据发送子步骤：请求建立 TCP 连接的虚拟机将数据报拷贝到域间通信通道的写出缓冲区中，通过事件通道通知目标虚拟机，目标虚拟机收到通知后从自身读入缓冲区中取出数据报，并释放读入缓冲区中的空间；通信双方重复上述过程，直至数据报发送完毕。

2. 如权利要求 1 所述的虚拟集群中的通信方法，其特征在于：

所述非域间通信步骤数据的传输子步骤中，所述数据报由报头和数据构成，报头用于在发送数据报的时候交换所需的控制信息，报头包含 8 个字段，从前到后依次为校验和、数据报序列号、数据起始索引值、数据负载值、数据报序列确认号、数据报索引确认号、数据报标识字段、通道号；

校验和字段存放对报头及数据所计算出的校验和的值；

数据报序列号字段用于标识数据报的先后顺序，第一个数据报序列号的值随机生成，之后每个数据报序列号值顺序加一；

数据起始索引值为所述数据在发送缓冲区中的起始索引值；

数据负载值为所述数据的字节数；

数据报序列确认号为下一个期待接收到数据报的序列号；

数据报索引确认号为接收缓存区中数据的结束索引值；

数据报标识字段存放是否要求确认的标识；

通道号字段为通道指针数组中项的索引。

3. 如权利要求 1 或 2 所述的虚拟集群中的通信方法，其特征在于：

所述非域间通信步骤中，数据传输子步骤包括下述过程：

A. 从发送缓冲区中根据网络最大传输单元的大小取出尽量多的数据，计算数据的校验和并将其存放在数据报报头校验和字段中，数据报报头中再依次填入数据报序列号、数据起始索引值、数据负载值、数据报确认序列号、数据报索引确认号、数据报标识字段、通道号字段的值，组装成一个数据报，使得数据报的大小不超过网络最大传输单元的大小；

B. 将组装成的数据报通过 Xen 的虚拟网络接口发送给通信对方；

C. 发送数据报个数每达到窗口值 W 的四分之一时，在最后一个数据报报头中加入要求确认标识，要求通信对方发送确认数据报；当已发送但未确认数据报个数达到窗口值 W 时，停止发送数据报，判断在时间 T 内是否收到确认数据报，是则继续发送，否则再次在一个数据报报头中加入要求确认标识，要求通信对方发送确认数据报；三倍时间 T 内仍未收到确认数据报，则转过程 A，重新组装未确认数据报；

D. 在发送完缓冲区中所有数据后，在最终数据报报头中加入要求确认标识，要求通信对方发送确认数据报；判断在时间 T 内是否收到确认数据报，是则结束，否则再次在最终数据报报头中加入要求确认标识，要求通信对方发送确认数据报；三倍时间 T 内仍未收到确认数据报，则转过程 A，重新组装未确认数据报。

一种虚拟集群中的通信方法

技术领域

[0001] 本发明属于虚拟化及集群计算领域，具体涉及一种虚拟集群中的通信方法。

背景技术

[0002] 近年来，以资源的高效组织、透明使用为目的的虚拟化技术快速发展，为计算机软硬件产业的发展提供了一个突破点。

[0003] 虚拟化是将底层物理设备与上层操作系统、软件分离的一种去耦合技术，它可以实现计算资源的高效灵活使用。计算系统虚拟化的实质就是针对个性化需求，高效组织计算资源，隔离具体的硬件体系结构和软件系统之间的紧密依赖关系，在动态环境中按需构建计算系统虚拟映像，构造可以适应用户需求的协同普适化任务执行环境，从而实现透明的可伸缩计算架构，提高计算资源的使用效率，发挥计算资源的聚合效能，使用户可以获得高效透明的服务。

[0004] 剑桥大学开发的虚拟机监视器 Xen 是一个开放源代码的虚拟机监视器，通过 Xen，多个操作系统可以同时在一台物理计算机上运行。运行在物理硬件和操作系统之间的虚拟机监视器 Xen 虚拟化了底层的物理硬件资源并且向运行在 Xen 上面的操作系统提供虚拟化的资源。运行在 Xen 上的操作系统被称为虚拟机，其中一个具有直接硬件访问特权的虚拟机被称为 Dom 0，其它不具有直接硬件访问特权的虚拟机被统称为 Dom U。

[0005] 在虚拟机管理器 Xen 中，提供了在虚拟机之间共享内存的机制 GrantTable（通过调用 Grant Table 提供的函数，一台虚拟机可以访问另一台虚拟机的内存）；还提供了事件通道机制用于一台物理计算机上的虚拟机之间传递消息；包含了用于存储虚拟机配置信息的内存区域 Xen Store（虚拟机管理器 Xen 中的一块特殊的内存，各虚拟机可以在其中写入信息和读取自己写入的信息，特权虚拟机 Dom 0 可以从中读取所有的信息）。

[0006] 虚拟集群包括多台处于运行状态的物理计算机，这些物理计算机通过高速的局域网络连接通信，每台物理计算机上运行有多个虚拟机，这些虚拟机上面运行着各种不同的任务，包括高性能科学计算、大规模分布式数据处理等。虚拟集群中的物理计算机包括一个或者多个物理 CPU（中央处理器），虚拟机监视器以及一台或多台虚拟机（指通过软件模拟的，具有完整硬件系统功能的，运行在一个完全隔离环境中的完整计算机系统），每台虚拟机上运行一个客户操作系统 (Guest OS)。虚拟机监视器采用灵活的处理器调度策略，以响应各个虚拟机不断变化的负载情况。

[0007] 在虚拟集群中，存在着三种类型的通信，虚拟集群内非域间通信（不同物理计算机上虚拟机之间的网络通信）、域间通信（同一个物理计算机上虚拟机之间的网络通信）和外部通信（虚拟集群中的虚拟机与外界计算机之间的通信）。

[0008] 在虚拟集群中，虚拟化的物理计算机之间通过高速网络相连，例如千兆以太网、VIA、Quadrics、Myrinet 和 InfiniBand（高速局域网络的几种类型）等，具有低延时、高带宽的特点。通常物理计算机之间的网络通信仍旧是通过复杂的传输控制协议和网际协议 (TCP/IP) 来进行通信的。TCP/IP 协议是为了适应 Internet 网络复杂的网络环境而设计

的，非常复杂，由于没有像 Internet 那样复杂的网络环境，所以 TCP/IP 协议并不太适合于虚拟集群计算，浪费了宝贵的计算资源。TCP/IP 协议是《计算机网络》课程的主要内容，见《TCP/IP 详解，卷 1- 卷 3》，史蒂文斯（美国）著，范建华等译，机械工业出版社。

[0009] 经过评测，在虚拟集群内非域间通信的情况下，测试表明 Linux 虚拟机的网络通信性能远低于物理 Linux 计算机系统的网络通信性能。网络数据接收性能会降低为物理 Linux 计算机系统的 1/2 到 1/3，网络数据发送性能会降低到物理 Linux 计算机系统的 1/5。

[0010] 在域间通信情况下，发送端虚拟机中的数据报需要通过 TCP/IP 协议栈、发送端虚拟机前端驱动 (Xen 虚拟网络接口驱动位于虚拟机中的一部分)，然后发送到位于 Dom 0(Xen 中具有硬件访问特权的虚拟机）中的后端驱动 (Xen 虚拟网络接口驱动位于 Dom 0 中的一部分)，经虚拟网桥（用于连接 Xen 中的各个虚拟机的后端驱动）转发到接收端虚拟机的前端驱动、TCP/IP 协议栈后才能达到应用程序，数据报穿越的长路径导致域间通信性能非常低下。

[0011] 如前所述，虚拟集群中虚拟机低效的网络通信性能会严重降低应用程序的效率，提高虚拟机网络通信的性能是一个亟待解决的问题。

发明内容

[0012] 本发明提出一种虚拟集群中的轻量级通信方法，解决虚拟化技术下的网络通信效率性能低下、非域间通信时虚拟集群中 TCP/IP 通信协议过于复杂、域间通信时数据报穿越的路径过长导致域间通信性能低下的问题，以提高虚拟集群网络通信效率。

[0013] 本发明在启动时，在虚拟机中创建一个虚拟字符设备，向该设备写 TCP 连接的套接字描述符时，连接建立子步骤利用该套接字描述符完成轻量级协议连接的建立；还在虚拟机的内存空间中分配通道指针数组，通道指针数组中的项指向轻量级通信协议的数据缓冲区，通道号是通道指针数组中项的索引，通过通道号可以方便地找到相应的轻量级通信协议数据缓冲区；同时会设置窗口值 W 和时间 T，窗口值 W 表示已发送但未确认的数据报的最大个数，时间 T 指的是等待数据报确认的最长时间。

[0014] 域间通信机制在相互通信的虚拟机之间开辟域间通信通道，域间通信通道包括读入缓冲区和写出缓冲区两个缓冲区和一个事件通道，读入和写出两个缓冲区分别负责接收和发送数据，事件通道用于相互通信的虚拟机之间传递消息。

[0015] 本发明发送的数据报和 IP 数据报一样，也有报头结构，用来在发送数据报的时候交换所需的控制信息。虚拟集群内非域间通信数据报的报头大小可以为 32 字节，包含 8 个字段，每个字段的大小可以为 4 个字节，从前到后分别为校验和、数据报序列号、数据起始索引值、数据负载值、数据报确认序列号、数据报索引确认号、数据报标识字段、通道号。

[0016] 本发明的一种虚拟集群中的通信方法，包括启动步骤、连接帮助步骤、非域间通信步骤、虚拟机发现步骤以及域间通信步骤，其特征在于：

[0017] (1) 启动步骤，为每台虚拟机创建虚拟字符设备，在每台虚拟机的内存中设置通道指针数组、窗口值 W 和时间 T，每台虚拟机在所在物理计算机的一块内存 Xen Store 中写入自己的虚拟机 ID 和 MAC 地址信息；所述窗口值 W 为 64 ～ 256 个，时间 T 为 0.1 ～ 1ms；

[0018] (2) 连接帮助步骤，包括下述子步骤：

[0019] (2.1)Linux 内核调用截获子步骤：请求建立 TCP 连接的虚拟机修改 Linux 内核代

码中的函数指针，截获上层应用程序对 TCP 连接建立函数的调用指令，根据调用指令中的目标 IP 地址判断网络通信是否属于虚拟集群内的非域间通信，是则请求建立一个用于建立轻量级通信连接时交换所需信息的 TCP 连接，转子步骤 (2.2)，否则转子步骤 (2.4)；

[0020] (2.2) 网络监听子步骤：TCP 连接的目标虚拟机接受 TCP 连接请求，把 TCP 连接套接字描述符写入到自身的虚拟字符设备中，转子步骤 (2.3)；

[0021] (2.3) 连接建立子步骤：写入 TCP 连接套接字描述符后，目标虚拟机在自身的内存中分配数据缓冲区并将其初始化，在通道指针数组中为该数据缓冲区分配通道号，然后通过 TCP 连接向请求建立 TCP 连接的虚拟机发送自身的 MAC 地址和通道号，接收对方的 MAC 地址和通道号，完成轻量级协议连接的建立，转步骤 (3)；

[0022] (2.4)TCP/IP 协议子步骤：通过 TCP/IP 协议的三次握手过程与目标地址建立起 TCP 连接，在传输数据时，TCP/IP 协议采用超时重传和捎带确认机制保证数据传输的可靠性，在流量控制上，TCP/IP 协议采用滑动窗口的方式，已发送且未确认的数据的字节数最大不能超过滑动窗口的大小，在 TCP/IP 协议完成数据报的处理后，转步骤 (4)；

[0023] (3) 非域间通信步骤，包括下述子步骤：

[0024] (3.1) 数据缓冲子步骤：请求建立 TCP 连接的虚拟机或者目标虚拟机发送数据时，将要发送的数据存放在自身数据缓冲区的发送缓冲区中，转子步骤 (3.2)；请求建立 TCP 连接的虚拟机或者目标虚拟机接收数据时，将接收到的数据存放在自身数据缓冲区的接收缓冲区中，等待应用程序读取；

[0025] (3.2) 数据传输子步骤：从发送缓冲区中取出数据组装成数据报，将数据报通过 Xen 的虚拟网络接口发送给对方虚拟机，直至所要发送的数据全部发送完毕；

[0026] (4) 虚拟机发现步骤，包括下述子步骤：

[0027] (4.1) 虚拟机扫描子步骤：请求建立 TCP 连接的虚拟机所在的物理计算机周期性地扫描自身内存的 Xen Store 区域，获取参与域间通信的虚拟机列表信息，转子步骤 (4.2)；

[0028] (4.2) 虚拟机信息广播子步骤：将参与域间通信的虚拟机列表信息组成列表信息数据报，通过网络广播到每一个参与域间通信的虚拟机中去，转步骤 (5)；

[0029] (5) 域间通信步骤，包括下述子步骤：

[0030] (5.1) 数据报截获子步骤：请求建立 TCP 连接的虚拟机收到列表信息数据报，将其中的虚拟机列表信息存放在自身内存中；然后截获经过 TCP/IP 协议处理的数据报，查看该数据报的目标地址是否包括在所述虚拟机列表信息中，是则转子步骤 (5.2)，否则将数据报通过 Xen 的虚拟网络接口发送给目标地址；

[0031] (5.2) 域间通信通道建立子步骤：请求建立 TCP 连接的虚拟机查看是否和目标虚拟机之间已经建立域间通信通道，是则转子步骤 (5.4)，否则转子步骤 (5.3)；

[0032] (5.3) 在自身内存中分配读入缓冲区和写出缓冲区并创建一个事件通道，然后发送通道建立数据报通知目标虚拟机将两个缓冲区的内存地址映射到目标虚拟机的内存地址空间中并且与所创建的事件通道绑定，目标虚拟机收到通道建立数据报后完成所述映射和绑定操作，转子步骤 (5.4)；

[0033] (5.4) 域间数据发送子步骤：请求建立 TCP 连接的虚拟机将数据报拷贝到域间通信通道的写出缓冲区中，通过事件通道通知目标虚拟机，目标虚拟机收到通知后从自身读

入缓冲区中取出数据报，并释放读入缓冲区中的空间；通信双方重复上述过程，直至数据报发送完毕。

[0034] 所述的虚拟集群中的通信方法，其特征在于：

[0035] 所述非域间通信步骤数据的传输子步骤中，所述数据报由报头和数据构成，报头用于在发送数据报的时候交换所需的控制信息，报头包含 8 个字段，从前到后依次为校验和、数据报序列号、数据起始索引值、数据负载值、数据报序列确认号、数据报索引确认号、数据报标识字段、通道号；

[0036] 校验和字段存放对报头及数据所计算出的校验和的值，用于保证传输的数据报的正确性；

[0037] 数据报序列号字段用于标识数据报的先后顺序，第一个数据报序列号的值随机生成，之后每个数据报序列号值顺序加一；

[0038] 数据起始索引值为所述数据在发送缓冲区中的起始索引值；

[0039] 数据负载值为所述数据的字节数；

[0040] 数据报序列确认号为下一个期待接收到数据报的序列号，用于对接收到的数据报进行确认；

[0041] 数据报索引确认号为接收缓存区中数据的结束索引值；

[0042] 数据报标识字段存放是否要求确认的标识；

[0043] 通道号字段为通道指针数组中项的索引，通过通道号可以方便地找到相应的轻量级通信协议数据缓冲区。

[0044] 所述的虚拟集群中的通信方法，其特征在于：

[0045] 所述非域间通信步骤中，数据传输子步骤包括下述过程：

[0046] A. 从发送缓冲区中根据网络最大传输单元 (MTU) 的大小取出尽量多的数据，计算数据的校验和并将其存放在数据报报头校验和字段中，数据报报头中再依次填入数据报序列号、数据起始索引值、数据负载值、数据报确认序列号、数据报索引确认号、数据报标识字段、通道号字段的值，组装成一个数据报，使得数据报的大小不超过网络最大传输单元的大小；

[0047] B. 将组装成的数据报通过 Xen 的虚拟网络接口发送给通信对方；

[0048] C. 发送数据报个数每达到窗口值 W 的四分之一时，在最后一个数据报报头中加入要求确认标识，要求通信对方发送确认数据报；当已发送但未确认数据报个数达到窗口值 W 时，停止发送数据报，判断在时间 T 内是否收到确认数据报，是则继续发送，否则再次在一个数据报报头中加入要求确认标识，要求通信对方发送确认数据报；三倍时间 T 内仍未收到确认数据报，则转过程 A，重新组装未确认数据报；

[0049] D. 在发送完缓冲区中所有数据后，在最终数据报报头中加入要求确认标识，要求通信对方发送确认数据报；判断在时间 T 内是否收到确认数据报，是则结束，否则再次在最终数据报报头中加入要求确认标识，要求通信对方发送确认数据报；三倍时间 T 内仍未收到确认数据报，则转过程 A，重新组装未确认数据报。

[0050] 本发明具有以下特点：

[0051] (1) 同时对域间通信和非域间通信进行优化

[0052] 本发明对非域间通信进行专门设计，避免了 TCP/IP 复杂的处理过程，降低了处理

器的处理负担，提高了网络通信的性能；对于域间通信，在相互通信的虚拟机之间共享内存开辟高速通信通道，缩短了数据报穿越的路径，大幅提高域间通信的性能；可以显著提高虚拟集群中网络通信的性能。

[0053] (2) 不依赖虚拟集群使用的具体网络硬件

[0054] 本发明的各个步骤可以由功能模块实现，与虚拟集群具体使用的网络硬件无关，不依赖于底层网络基础设施。

[0055] (3) 原有二进制程序可以直接运行于本发明

[0056] 本发明保持了应用程序的套接字函数接口不变，具有对已有应用程序的二进制兼容性，原有的应用程序不用重新编写或是重新编译就可以在本发明运行，提高现有应用的网络通信的性能。

附图说明

[0057] 图 1 为本发明的工作流程图；

[0058] 图 2 为非域间通信步骤的数据传输子步骤工作流程图。

具体实施方式

[0059] 为了验证本发明的可行性和有效性，在真实环境下本发明的步骤可编写为计算机程序，部署到虚拟集群环境中，并和没有采用本发明时虚拟集群内虚拟机非域间通信和域间通信的性能进行比较。

[0060] 本实施例共采用两台硬件配置相同的计算机，两台计算机的编号分别为 PC1 和 PC2，其配置如表 1 所示。

[0061] 表 1：实验配置环境

[0062]

编号 / 配置	PC1	PC2
CPU	Pentium(R) Dual-Core CPU E5200, 2.50GHz	Pentium(R) Dual-Core CPU E5200, 2.50GHz
内存	2GB	2GB
硬盘	190GB	190GB
网络接口	千兆以太网卡	千兆以太网卡
操作系统	Red Hat Linux 5.3	Red Hat Linux 5.3
虚拟机管理器	Xen 3.2	Xen 3.2

[0063] 在 PC1 上启动两台虚拟机，标号分别为 VM1 和 VM2，在 PC2 上启动一台虚拟机，编号为 VM3。

[0064] 在未采用本发明的情况下，利用 netperf 程序（一个用于测试网络吞吐量的测试程序）测试 VM1 和 VM3 之间 TCP 协议非域间通信的性能，利用 netperf 程序测试 VM1 和 VM2 之间 TCP 协议域间通信的性能。

[0065] 将本发明部署到测试环境中，启动时，为 VM1、VM2 和 VM3 中创建虚拟字符设备，在 VM1、VM2 和 VM3 的内存中分配通道指针数组，设定窗口值 W 的大小为 128、时间 T 的大小为 0.5ms，并且将虚拟机 VM1、VM2 的 ID 号和 MAC 地址写到 PC1 的 Xen Store 内存区域中，将 VM3 的 ID 号和 MAC 地址写到 PC2 的 Xen Store 内存区域中。

[0066] 在 VM1 和 VM3 之间用 netperf 程序测试非域间通信的性能。VM1 向 VM3 请求建立 TCP 连接时，Linux 内核截获子步骤判断出该 TCP 连接请求目标地址属于虚拟集群内的非域间通信，在通信对方通过网络监听子步骤的帮助下，连接建立子步骤完成轻量级协议连接的建立；当 VM1 发送数据到 VM2 时，数据缓冲子步骤将发送的数据缓冲在发送缓冲区中，由数据传输子步骤负责数据的发送，当 VM1 收到 VM2 发来的数据时，将接收到的数据缓冲在接收缓冲区中，等待应用程序从接收缓冲区中读取数据。

[0067] 在 VM1 和 VM2 之间用 netperf 程序测试域间通信的性能。VM1 向 VM3 请求建立 TCP 连接时，Linux 内核截获子步骤判断出该 TCP 连接请求目标地址不属于虚拟集群内的非域间通信，转由 TCP 协议子步骤处理；虚拟机发现子步骤从物理计算机 Xen Store 内存中获取虚拟机列表信息并且通过列表信息数据报广播到 VM1 和 VM2 中去；当数据报截获子步骤截获到目标地址为 VM2 的数据报时，查看 VM1 和 VM2 之间是否建立了域间通信通道，是则将数据报拷贝到域间通信通道中向 VM2 传递数据的缓冲区中，并通过事件通道通知 VM2 读取数据报，否则由域间通信通道建立子步骤在 VM1 和 VM2 间建立域间通信通道，然后则将数据报拷贝到域间通信通道中向 VM2 传递数据的缓冲区中，并通过事件通道通知 VM2 读取数据报。

[0068] 测试结果如表 2 所示。

[0069] 表 2：虚拟集群内非域间通信和域间通信吞吐量测试结果

[0070]

测试类型	虚拟集群内非域间通信 (Mb/s)	域间通信 (Mb/s)
TCP 协议测试结果	748.53	2489.25
本发明测试结果	767.99	6397.37
性能提高幅度	2.6%	157%

[0071] 测试结果表明，在虚拟集群内非域间通信的情况下，本发明的吞吐量较 TCP 协议提高约 2.6%；在域间通信的情况下，本发明的吞吐量较 TCP 协议提高约 157%。

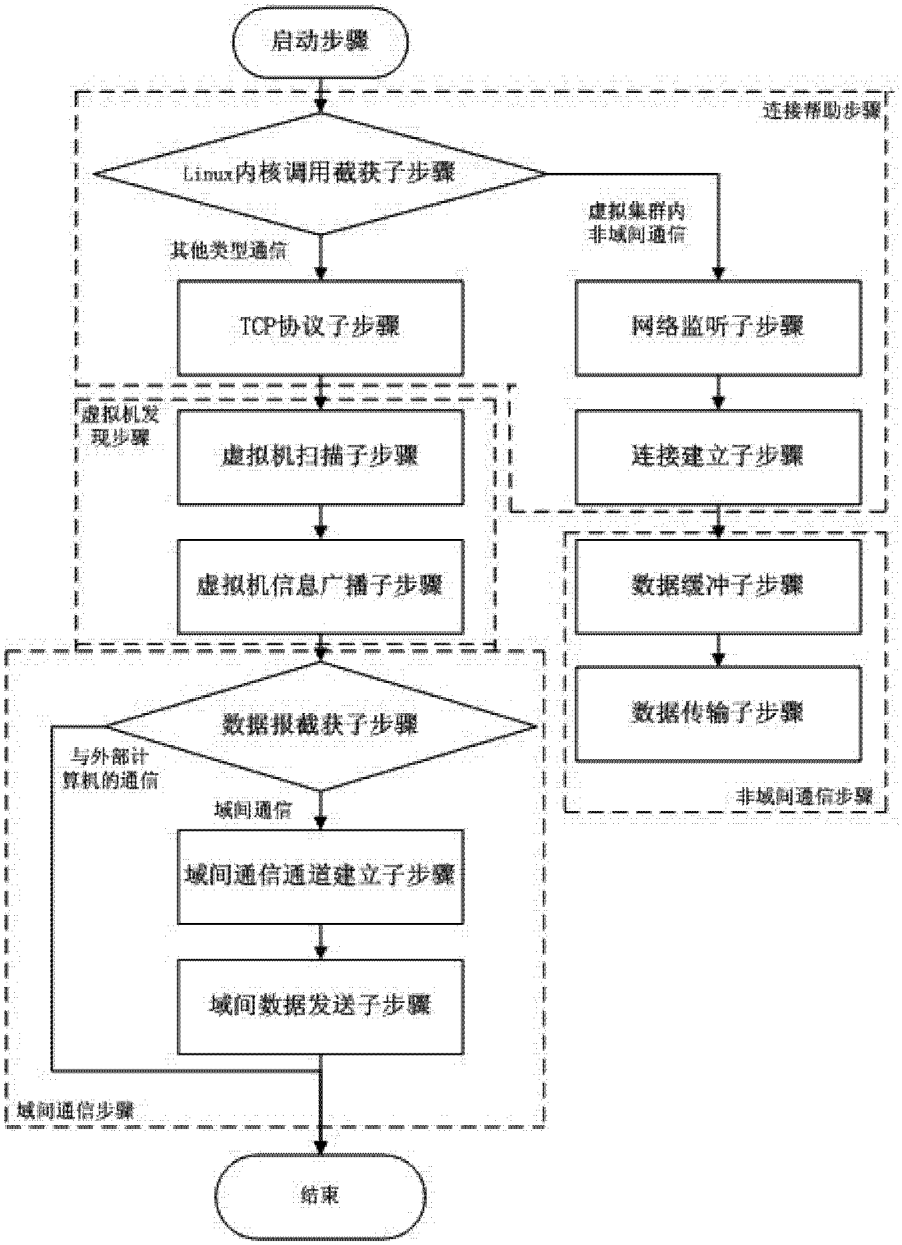

启动步骤
连接帮助步骤
Linux内核调用截获子步骤
虚拟集群内
非域间通信
其他类型通信
TCP协议子步骤
网络监听子步骤
虚拟机发
现步骤
虚拟机扫描子步骤
连接建立子步骤
虚拟机信息广播子步骤
数据缓冲子步骤
数据传输子步骤
非域间通信步骤
数据报截获子步骤
与外部计
算机的通信
域间通信
域间通信通道建立子步骤
域间数据发送子步骤
域间通信步骤
结束

图 1

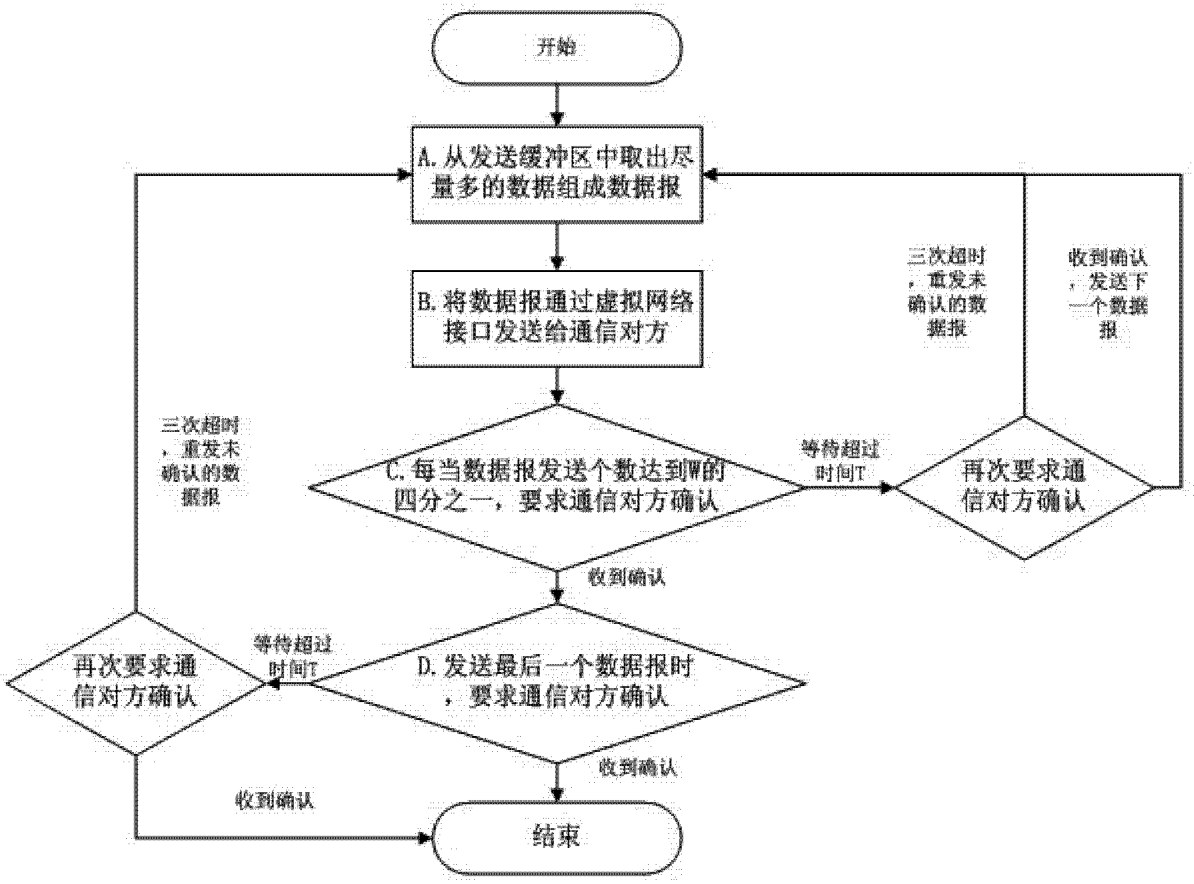

开始
A. 从发送缓冲区中取出尽量多的数据组成数据报
B. 将数据报通过虚拟网络接口发送给通信对方
C. 每当数据报发送个数达到W的四分之一，要求通信对方确认
三次超时，重发未确认的数据报
等待超过时间T
再次要求通信对方确认
三次超时，重发未确认的数据报
收到确认，发送下一个数据报
收到确认
D. 发送最后一个数据报时，要求通信对方确认
等待超过时间T
再次要求通信对方确认
收到确认
收到确认
结束

图 2